

Serial No. 10/631,062

1. (Currently Amended) A method in a data processing system for handling exception vectors by firmware, the method comprising:
  - identifying an exception vector to form an identified exception vector when control is passed from an operating system to the firmware, wherein the exception vector contains objects for processing traps and interrupts for the operating system;
  - saving the identified exception vector to form a saved exception vector;
  - replacing the identified exception vector with a substitute vector;
  - restoring the saved exception vector when control is returned to the operating system to form a restored exception vector, wherein the restored exception vector continues execution; and
  - responsive to a processor, other than an associated processor associated with the ~~save~~ ~~interrupt vector~~ saved exception vector, generating an error for the identified exception vector replaced by the substitute vector, placing the processor in a slave loop until the saved exception vector is restored, wherein context for the processor in the slave loop is stored to form a stored context, wherein the processor and the associated processor are identified using processor identification numbers, and wherein the placing step is conditionally performed only if a processor identification number for the processor does not match an associated processor identification number for the associated processor;
  - wherein the data processing system is a symmetric multi-processor data processing system having a plurality of logical partitions with each of the plurality of logical partitions (i) having at least one physical processor assigned thereto, and (ii) executing its own operating system which is a different instance from other operating systems executing on other of the plurality of logical partitions.
2. (Previously Presented) The method of claim 1 wherein the saving, replacing, and restoring steps are conditionally performed only if the data processing system is in a debug mode as determined by a debug flag maintained by the firmware.
- 3-4. (Cancelled)
5. (Previously Presented) The method of claim 1, wherein the processor is restored by restoring the context using the stored context and releasing the processor to the restored exception vector.
6. (Cancelled)

Serial No. 10/631,062

7. (Currently Amended) A method operable by firmware in a data processing system for managing exception vectors, the method comprising:
- receiving control from an operating system;
  - replacing, by the firmware, an exception vector with substitute code, wherein the exception vector contains objects for processing traps and interrupts for the operating system; and
  - restoring, by the firmware, the exception vector when control is returned to the operating system, wherein processors, other than a particular processor creating the exception vector, encountering the substitute code are suspended by the firmware until control is returned to the operating system, wherein the processors and the particular processor are identified using processor identification numbers, and wherein the suspension of individual ones of the processors is conditionally performed only if a processor identification number for a given processor of the processors does not match a particular processor identification number for the particular processor;
- wherein the data processing system is a symmetric multi-processor data processing system having a plurality of logical partitions with each of the plurality of logical partitions (i) having at least one physical processor assigned thereto, and (ii) executing its own operating system which is a different instance from other operating systems executing on other of the plurality of logical partitions.
8. (Previously presented) The method of claim 7, wherein the processors are suspended by placing the processors in a slave loop by the firmware only if a debug mode for the firmware is enabled.
9. (Currently Amended) A data processing system for handling exception vectors by firmware, the data processing system comprising:
- identifying means for identifying an exception vector to form an identified exception vector when control is passed from an operating system to the firmware, wherein the exception vector contains objects for processing traps and interrupts for the operating system;
  - saving means for saving the identified exception vector to form a saved exception vector;
  - replacing means for replacing the identified exception vector with a substitute vector;
  - restoring means for restoring the saved exception vector when control is returned to the operating system to form a restored exception vector, wherein the restored exception vector continues execution; and

Serial No. 10/631,062

means, responsive to a processor, other than an associated processor associated with the ~~save interrupt vector~~ saved exception vector, generating an error for the identified exception vector replaced by the substitute vector, for placing the processor in a slave loop until the saved exception vector is restored, wherein context for the processor in the slave loop is stored to form a stored context, wherein the processor and the associated processor are identified using processor identification numbers, and wherein the means for placing the processor in the slave loop is conditionally invoked only if a processor identification number for the processor does not match an associated processor identification number for the associated processor;

wherein the data processing system is a symmetric multi-processor data processing system having a plurality of logical partitions with each of the plurality of logical partitions (i) having at least one physical processor assigned thereto, and (ii) executing its own operating system which is a different instance from other operating systems executing on other of the plurality of logical partitions.

10. (Previously Presented) The data processing system of claim 9 further comprising:  
means for conditionally invoking the saving means, replacing means, and restoring means only if the data processing system is in a debug mode as determined by a debug flag maintained by the firmware.

11-12. (Cancelled)

13. (Previously Presented) The data processing system of claim 9, wherein the processor is restored by restoring the context using the stored context and releasing the processor to the restored exception vector.

14. (Cancelled)

15. (Currently Amended) A multi-partitioned data processing system for managing exception vectors, the data processing system comprising:  
receiving means for receiving control from an operating system;  
replacing means for replacing an exception vector with substitute code, wherein the exception vector contains objects for processing traps and interrupts for the operating system; and  
restoring means for restoring the exception vector when control is returned to the operating system, wherein processors, other than a particular processor creating the exception

Serial No. 10/631,062

vector, encountering the substitute code are suspended until control is returned to the operating system, wherein the processors and the particular processor are identified using processor identification numbers, and wherein the suspension of individual ones of the processors is conditionally performed only if a processor identification number for a given processor of the processors does not match a particular processor identification number for the particular processor;

wherein the multi-partitioned data processing system comprises a plurality of logical partitions with each of the plurality of logical partitions (i) having at least one physical processor assigned thereto, (ii) executing its own operation system which is a different instance from other operating systems executing on other of the plurality of logical partitions, and (iii) executing its own partition firmware which is loaded into the each of the plurality of logical partitions when the partitions are instantiated; and

wherein the receiving means, the replacing means and the restoring means are each performed by the partition firmware of a given partition.

16. (Previously Presented) The data processing system of claim 15, wherein the processors are suspended by placing the processors in a slave loop by the partition firmware of the given partition.

17. (Currently Amended) A computer program product in a computer recordable-type medium and operable with a data processing system for handling exception vectors by firmware, the computer program product comprising:

first instructions for identifying an exception vector to form an identified exception vector when control is passed from an operating system to the firmware, wherein the exception vector contains objects for processing traps and interrupts for the operating system;

second instructions for saving the identified exception vector to form a saved exception vector;

third instructions for replacing the identified exception vector with a substitute vector;  
[[and]]

fourth instructions for restoring the saved exception vector when control is returned to the operating system to form a restored exception vector, wherein the restored exception vector continues execution; and

fifth instructions, responsive to a processor, other than an associated processor associated with the saved exception vector, generating an error for the identified exception vector replaced

Serial No. 10/631,062

by the substitute vector, for placing the processor in a slave loop until the saved exception vector is restored, wherein the processor and the associated processor are identified using processor identification numbers, and wherein the placing step is conditionally performed only if a processor identification number for the processor does not match an associated processor identification number for the associated processor;

wherein the data processing system is a symmetric multi-processor data processing system having a plurality of logical partitions with each of the plurality of logical partitions (i) having at least one physical processor assigned thereto, and (ii) executing its own operating system which is a different instance from other operating systems executing on other of the plurality of logical partitions.

18. (Previously presented) The computer program product of claim 17 ~~further comprising:~~  
~~fifth instructions for generating an error for the identified exception vector replaced by the substitute vector in response to a processor, other than an associated processor associated with the save interrupt vector, and placing the processor in a slave loop until the saved exception vector is restored~~ wherein the second instructions for saving, the third instructions for replacing, and the fourth instructions for restoring are conditionally performed only if the data processing system is in a debug mode as determined by a debug flag maintained by the firmware.

19. (Cancelled)

20. (Previously presented) The computer program product of claim 18, wherein context for the processor in the slave loop is stored to form a stored context.

21. (Previously Presented) The computer program product of claim 20, wherein the processor is restored by restoring the context using the stored context and releasing the processor to the restored exception vector.

22. (Cancelled)

23. (Currently Amended) A computer program product in a computer recordable-type medium and operable with a data processing system for managing exception vectors, the computer program product comprising:

first instructions for receiving control from an operating system;



Serial No. 10/631,062

second instructions for replacing an exception vector with substitute code, wherein the exception vector contains objects for processing traps and interrupts for the operating system; and

third instructions for restoring the exception vector when control is returned to the operating system, wherein processors, other than a particular processor creating the exception vector, encountering the substitute code are suspended until control is returned to the operating system, wherein the substitute code provides particular debug code for debugging the computer program product, wherein the processors and the particular processor are identified using processor identification numbers, and wherein the suspension of individual ones of the processors is conditionally performed only if a processor identification number for a given processor of the processors does not match a particular processor identification number for the particular processor;

wherein the data processing system is a symmetric multi-processor data processing system having a plurality of logical partitions with each of the plurality of logical partitions (i) having at least one physical processor assigned thereto, and (ii) executing its own operating system which is a different instance from other operating systems executing on other of the plurality of logical partitions.

24. (Previously Presented) The computer program product of claim 23, wherein the processors are suspended by placing the processors in a slave loop by the computer program product only if a debug mode for the computer program product is enabled.